交通部中央氣象局委託研究計畫成果報告

Developments of An Artifical Neural Network Model and Its Application for Long-range Rainfall Prediction in Taiwan

計畫類別：□國內　■國外
計畫編號：MOTC-CWB-90-2M-12
執行期間：90 年 2 月 1 日至 90 年 12 月 31 日
計畫主持人：朱寶信
執行單位：Climate System Enterprise

中華民國九十年十二月

Developments of An Artificial Neural Network Model and Its Application for Long-range Rainfall Prediction in Taiwan

Final Report to the Central Weather Bureau
Under Contract Number CWB-90-2M-12

Pao-Shin Chu, Ph.D.
Climate System Enterprise
Honolulu, Hawaii, U.S.A.

December 28, 2001

1. Introduction

Interests in the artificial neural network (NN) model and its applications in climate science are growing rapidly during the last decade. These interests result mainly from progress in developments of efficient training algorithms in the NN field, the accessibility of large quantities of climate data for analysis, improved understanding of the complex climate systems, and/or the availability of powerful personal computer hardware for running simulations and predictions of NN models. In contrast to the conventional linear, deterministic models, the NN is a powerful and nonlinear system capable of handling a wide range of applications. These include seasonal climate predictions (e.g., Hastenrath, 1995; Sahai et a., 2000; Chu and Yan, 2000), statistical downscaling (e.g., Chu and Wu, 2001), and time series analysis of nonlinear dynamic systems. A NN system can be recognized as an advanced pattern recognition technique, and this pattern recognition can be employed efficiently for simulations and predictions.

The main objective of this project is to develop an artificial NN model and test this model's ability in predicting summer rainfall for Taiwan. This is the second year of the project. During the first year, we developed a prototype NN model using the gradient descent method which is a slow but stable learning law. For the second year, our focus is to improve the model's performance by introducing a faster learning law called the adaptive learning rate. To allow the learning be as fast as possible, an appropriate weight initialization method is introduced. With these improvements in model's algorithms, we have tested model's forecasting skills using cross-validation techniques. The computer code for running NN models has been successfully transferred to the Forecast Center of Central Weather Bureau. To complement the current project, we have applied the NN model to simulate daily rainfall in Taiwan (Please see Chu and Wu, 2001 in the attachment).

2. NN structure

The network consists of three layers: the input layer, the hidden layer, and the output layer (Fig. 1). It is fully connected so that there are links between all the nodes in the adjacent layer. Each link has a connection strength, called the weight, which is stored by the neuron at the receiving end of the link. The weight is unknown and needs to be determined iteratively. Note that the link is separate from each input node to the hidden node, and from the hidden node to the output node.

3. Basic concepts of NN systems

As demonstrated in Fig. 1, each node in the input layer brings into the network the value of one independent variable (X_i). In our case, X_i may represent each dominant extended empirical orthogonal function (EEOF) mode of the SST indices. Z_k may represent simulated rainfall outputs at each station and they need to be compared with the actual rainfall values for the purpose of determining model's ability in predictions or simulations. Each hidden node calculates a weighted sum of the input using a sigmoid function, which simply squashes the sum down to a limited range, say, between 0 and 1. In this project, we used a logistic function (g(u)) and hyperbolic tangent function (tanh(u)) such as:

$$g(u) = \frac{1}{1 + e^{-u}}$$

$$\tanh(u) = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$

where u is the weighted sum of the value received from the hidden node. Each output node performs a similar function as hidden node by calculating the weighted sum from the output of hidden layers. To simplify the computation and understanding the network, the output node uses a linear function (Hsieh and Tang, 1998). The output values (z) are then compared with the target value (t) to determine the error (E), which is shown as:

$$E = \sum_{k=1}^{K} (z_k - t_k)^2$$

where K is the number of output nodes.

For each observation used as an input, there is a forward pass through the network, from inputs to outputs. Each output node then propagates the errors back to the hidden nodes (i.e., backpropagation). The hidden nodes use these errors to determine which direction and how much the weights should be changed. The backpropagation is designed to minimize the average squared errors. The key concept in backpropagation is the sensitivity of the network's error to changes in its weights. In other words, we need to know how the partial derivatives of the error are with respect to the weights. Training begins with arbitrary values of the weights and proceeds iteratively. Each iteration is called an epoch. In each epoch, the network adjusts the weight in the direction that reduces the error. As iteration continues, the weights generally converge to the optimal values. The overall structure is as follow.

```
Do
    For i = 1 to n
        Go to subroutine Forward
        Go to subroutine Backward
    Continue
    Go to subroutine Changeweights
Enddo
```

The method to calculate the weight changes is gradient descent, which is a conventional way. Simply stated, one selects an arbitrary point in weight space and computes the slope of the error surface at this point. Change the weights in the direction in which the error surface goes down most steeply (i.e., steepest descent). Continue repeating the process until the new weight approaches the lowest point in the error surface. Steepest descent involves moving a small step down the local gradient of a scalar field. As an example, the gradient for the output layer is:

$$\delta_k = d\delta_k \cdot \ (z_k - t_k)^2$$

where $d\delta_k$ is the derivative of the sigmoid function at z_k for each output node k.

In comparison to the steepest descent, the method of Adaptive Learning Rates (ALR) is much faster to converge on the set of weights that produce minimum errors. The ALR is also very dependable, not prone to get into troubles. Although it requires the knowledge of some parameters, the results are not sensitive to the values of these parameters (Smith, 1996). Because it is faster, dependable, and highly automatic, we recently adopted this method. The idea of ALR is simple. Let e denote different learning rate for each weight in the network. If the direction in which the error decreases at this weight-change is the same as it has been decreasing recently, one makes e larger. Otherwise, makes e smaller. Once the learning rate is determined, the actual weight-change is given as:

$$c_m = -e_m \, d_m$$

where d_m is the partial derivative of the error at a weight at epoch m.

To begin the learning process, we need to assign some initial values to the weights. Note that initial values were fixed and the learning rate was slow during the first year of the project. Recently, an appropriate weight initialization scheme has been implemented so the learning becomes as fast as possible. When the weighted sum of inputs is close to zero, a node's output is close to 0.5, which is the midpoint of the logistic function. In the face of uncertainty, this mid-range value would be the best guess.

For the hidden nodes, the initial weights can be assigned to a very small value so that a node produces mid-range outputs. However, these weights should be different from one node to another. For the output nodes, a different strategy is adopted. The initial weights should be large since these weights are used in calculating the error derivatives of the hidden-node weights. It is recommended to keep half of the initial weights with the value of one and the other half with the value of minus one (Smith, 1996).

4. Application

We have collected, processed, and updated long-term rainfall and sea surface temperature (SST) records for this project. Rainfall records for 16 major stations in Taiwan during the period 1956-2000 were kindly provided by Mr. Guay-Hong Chen of the Forecast Center of the Central Weather Bureau. Monthly rainfall data were summed to seasonal values (July, August, and September) before being standardized. Likewise, the monthly SST data are transformed into standardized seasonal anomalies. For the SST used as the predictor variable, the domain covers the Pacific Ocean (50°N-40°S, 120°E-90°W) and the Indian Ocean (20°N-40°S, 20°E-120°E). This is different from Chu (1998) in which only the Pacific SSTs were used. To facilitate the analyses, the original SST data, which are at 2.5° latitudes x 2.5° longitudes, are averaged into 10° lat by 10° long boxes. This yields 125 boxes for the Pacific Ocean and 41 boxes for the Indian Ocean. By including the Pacific Ocean and the Indian Ocean, it is possible to test the importance of near-global boundary conditions on summer rainfall predictability for Taiwan.

Another new approach in this project is the consideration of the well-known persistence in SSTs. We calculated the Extended Empirical Orthogonal Function (EEOF) of SSTs at 3-month intervals over a one-year period prior to the summer rainy season. That is, the SST field evolves over one year period. For instance, at one-month lead, the SST from June of the previous year to May of the current year over the Pacific and Indian

Oceans are used. The leading six EEOF modes are retained because they are significant from the background noise (Wilks, 1995). In addition to spatial compression, the EEOF analysis enables us to capture propagating features of the SST. This is achieved by stacking the four temporal series of the SST field into a large matrix so that the evolution of the SST spatial pattern over a one year period is preserved. Note that the evolutionary feature of the SST was not considered in Chu (1998) for predicting Mei-yu rainfall in Taiwan.

To evaluate the overall forecasting ability of the NN system, a cross-validation technique is used. Cross-validation is a computer-intensive method. Each time, one point from the rainfall records and SST datasets is omitted, the NN model is reconstructed, and a forecast (or hindcast) for the omitted case (rainfall) is made. By repeatedly going through the entire records, one may obtain as many numbers of forecasts as the original observations. The hit rate is used to provide a verification measure, and both the observed and predicted rainfall is expressed by their percentile rank. A 3 x 3 contingency table of categorical forecast is used (Chu and Wu, 2001), where dry condition refers to <33.3%, normal condition is between 33.3% and 66.6%, and wet condition is more than 66.6%. The hit rate is defined as the ratio of number of hit (both forecast and observation fall into the same category) to the total number of events. Note that hit rate varies between zero and one, with zero being the worst skill and one being the perfect skill.

As an example, Fig. 2 shows the cross-validated forecast skill of the NN model in summer rainfall prediction at one-month lead. A logistic function is used. Evidently, 15 out of all 16 stations exhibit a moderate hit rate with a value greater than 0.4. The higher skill is found at Keelung (0.59), Jiu-Yueh-Tan (0.52), Alishan (0.50), and Hsinchu (0.50). The average rate of these 16 stations is 0.443. In a comparison study, an advanced linear model called canonical correlation analysis (CCA) was developed in predicting JAS rainfall using the same predictor field and same EEOF modes as the NN prediction system (Chu and Yan, 2000). The average rate of CCA-based prediction system turns out to be 0.386, lower than the NN model. If a hyperbolic tangent function is used in the NN system, the spatial distribution of hit rate varies somewhat (not shown), and the best skill is found in Penghu (0.69), and Hengchun (0.67). This result is interesting since Penghu always suffers from drought and the high predictability of summer rainfall demonstrated in this project may pave a way to better cope future climate uncertainty for an islet in the Taiwan Strait. There is a slight decrease in predictive skill for Keelung (0.54) and Hsinchu (0.48) in comparison to that in Fig. 2. The average rate from 16 stations based on the hyperbolic tangent function is 0.427, a value slightly lower than the case when the sigmoid function was employed.

5. Summary and future work

During the two-year project period (2000-2001), a great amount of effort has been devoted to the development of an state-of-the-art computer code of neural networks and their applications to summer rainfall prediction as well as statistical downscaling of daily rainfall in Taiwan (Chu and Wu, 2001). For the first year, a forward computer program and an error derivative algorithm were developed, and a linkage of various subroutines into a comprehensive computer program was made. For the second year, a more efficient algorithm called the adaptive learning rate to achieve minimum errors is implemented.

This method is faster and more reliable than the commonly used steepest descent method. In order for the learning process to start as fast as possible, a weight initialization procedure is introduced. This overcomes the problem of the fixed initial values for weights during the first year. Moreover, given these additions, the NN model's performance in rainfall prediction has been tested in a cross-validation scheme. The predictand is summer rainfall for 16 stations in Taiwan and the predictor field is the extended EOF of SST from the Pacific and Indian Oceans of the preceding seasons. Prediction experiment is conducted at a one-month lead. Forecasted rainfall is compared with the observed rainfall and hit rate is used to provide a measure of predictive skills. Results indicate that the cross-validated hit rate for most stations is well above the no skill level and a few stations have values greater than 0.5. Thus, there is a moderate predictability for summer monsoon rainfall, given the knowledge of the antecedent SST pattern and its time evolution. Also note that the overall predictive skill by the NN model is better than that by the corresponding, linear CCA model.

It should be noted that only the SST is used as the predictor variable. To the extent that the Earth's climate system we wish to predict is complex, predictive skills exhibited in this project can be regarded as a lower bound estimate for the actual predictability of the system. To improve the skill, other variables such as atmospheric teleconnection indices and the prior state of seasonal rainfall may be included. Furthermore, one may attempt to optimize the weights of the predictor fields prior to the prediction experiments. Recently, in predicting seasonal rainfall for east Africa, Ntale and Gan (2001) demonstrated an overall improvement in predictive skills relative to those which are unoptimized.

References

Chu, P.-S., 1998: Short-term climate prediction of Mei-yu rainfall for Taiwan using CCA. Int. J. Climatol., 18, 215-224.

Chu, P.-S., and W. Yan, 2000: Short-term climate predictability of summer rainfall for Taiwan. Preprints, 2nd International Ocean and Atmosphere Conf. Taipei, Taiwan, 10-12 July, 53-58.

Chu, P.-S., and M.-C. Wu, 2001: Regional precipitation simulations for Taiwan using an artificial neural network downscaling approach. Submitted for publication.

Hastenrath, S., L. Greischar, and J.L. McClelland, 1995: Prediction of summer rainfall over South Africa. J. Climate, 8, 1501-1508.

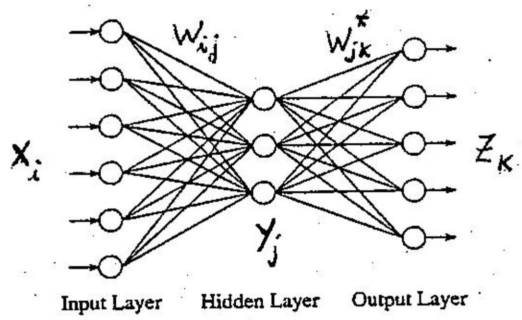
Hsieh, WW., and B. Tang, 1998: Applying neural network models to prediction and data analysis in meteorology and oceanography. Bull. Amer. Meteor. Soc., 79, 1855-1868.

Ntale, H.K., and T.Y. Gan, 2001: Prediction of East African seasonal rainfall using simplex-canonical correlation analysis. J. Climate, accepted for publications.

Sahai, A.K., Soman, M.K., and V. Satyan, 2000: All India summer monsoon rainfall prediction using an artificial neural network. Cli. Dyn. 16, 291-302.

Smith, M., 1996: Neural networks for statistical modeling. International Thomson Computer Press, Boston.

Wilks, D.S., 1995: Statistical methods in the atmospheric sciences. Academic Press, 467 pp.



Neural network

Fig. 1. A three-layer feed-forward neural network model. In this example, there are six nodes in the input layer, three nodes in the hidden layer, and five neurons in the output layer. The w_{ij} and w^*_{jk} are the weights.

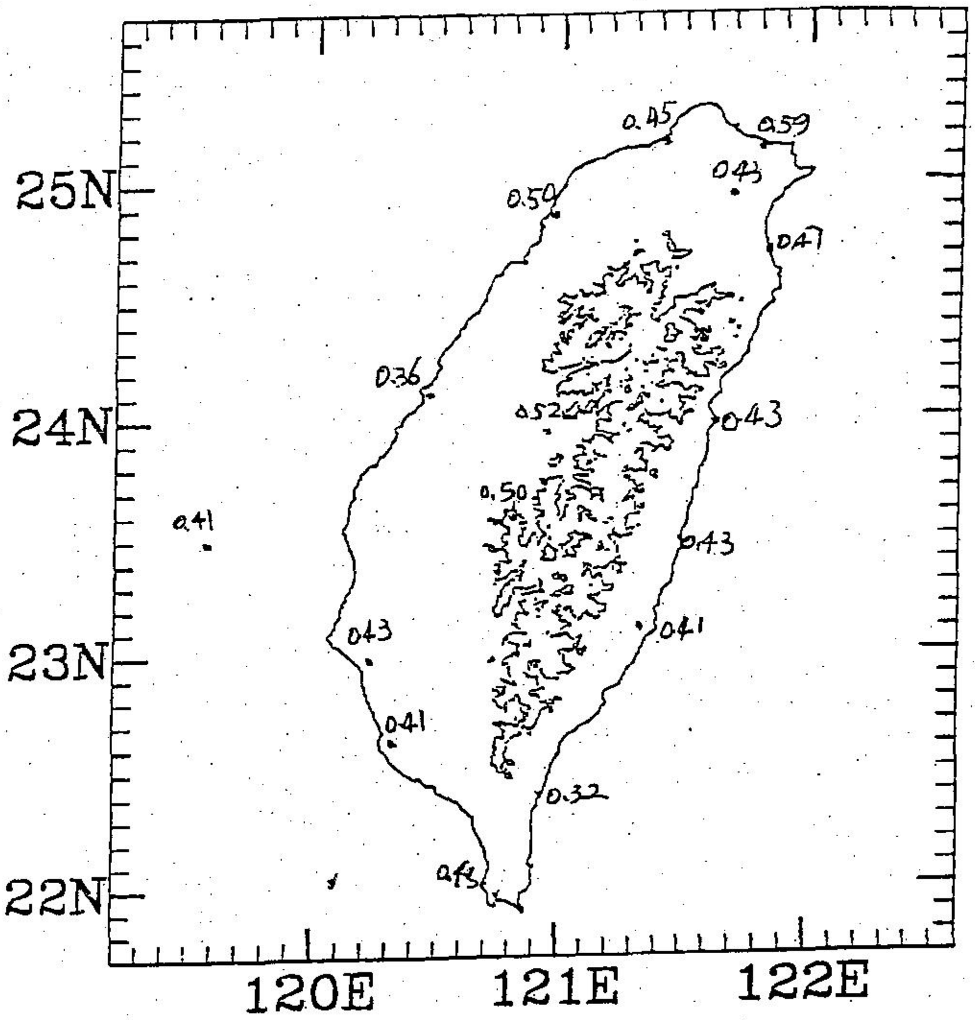


Fig. 2. Cross-validated hit rate of NN prediction for JAS rainfall during 1956-2000 in Taiwan. The lead is one month and the logistic function is used.